Claims

We Claim:

1. A computer implemented method for monitoring up-stream and down-stream software applications in a message queuing transmission system with at least one processing task running, the method comprising the steps of:

forming a first queue group by assigning an input queue group identifier to a first input queue and to a second input queue within a message queuing transmission system;

forming a second queue group by assigning an output queue group identifier to a first output queue and to a second output queue within the message queuing transmission system;

assigning a first queue identifier to the first input queue, a second queue identifier to the second input queue, a third queue identifier to the first output queue, and a fourth queue identifier to the second output queue;

assigning a task identifier to a processing task within the message queuing transmission system;

determining a first number of messages stored in the first input queue, a second number of messages stored in the second input queue, a third number of messages stored in the first output queue, and a fourth number of messages stored in the second output queue;

determining activation status of the processing task; and

gathering, in a task monitor storage area, the first number of messages stored in the first input queue, the second number of messages stored in the second input queue, the third number of messages stored in the first output queue, the fourth number of messages stored in the second output queue, and the activation status of the processing task.

2. The method of claim 1, repeating the steps of determining and the step of gathering according to a refresh time interval of a refresh counter.

3. The method of claim 1, further including the step of presenting contents of the task monitor storage area on a display screen.

4. The method of claim 1, further including the step of writing an error message from the processing task to an error-log queue.

5. The method of claim 4, further including the step of presenting the error message on a display screen.

6. The method of claim 1, wherein the first queue identifier, the second queue identifier, the third queue identifier, and the fourth queue identifier are each different.

7. The method of claim 1, wherein the processing task is a background task.

8. The method of claim 1, further including the steps of:

storing the task identifier in a task identifier table;

storing the first queue group identifier and the second queue group identifier in a queues groups table; and storing the first queue identifier, the second queue identifier, the third queue identifier, and the fourth queue identifier in a queues table.

9. The method of claim 1, further including the step of assigning a reply-to-group identifier to a reply queue for receiving a reply message generated in response to an outgoing message, wherein the reply message is generated by a down-stream software application for receiving the outgoing message.

10. The method of claim 9, wherein the reply message is taken from the reply queue and processed by a reply task.

11. The method of claim 9, further including the step of computing a time interval between writing the outgoing message to an output queue of the message queuing transmission system and writing the reply message to the reply queue.

12. The method of claim 11, further including the step of comparing said time interval with a predetermined time control interval.